

YTM32B1MD2x HCU Demo 使用说明

简述

YTM32B1MD2x 支持非对称算法 ECC-192/256/384 以及 RSA-1024/2048/3072/4096

特封出静态库，下面给出个简单示例

Demo 中展示

- AES-ECB 使用软件密钥进行加密
- AES-CMAC 使用软件密钥进行签名与验签，使用硬件密钥进行签名与验签
- SHA-256 的签名与验签
- RSA-1024 使用软件密钥进行加密与解密
- RSA-4096 使用硬件密钥进行加密与解密
- ECC-256 的基础运算，点加与点乘运算
- ECDSA 公钥生成，签名生成，以及签名认证

发布清单

- HCU demo 应用程序 (双击 HCU_Demo.yct 生成工程)
- lib_hcu.a (HCU 的驱动包，适用于 GCC 与 YuntulDE 开发环境)
- mdk_hcu.lib (HCU 的驱动包，适用于 KEIL 开发环境)
- iar_hcu.a (HCU 的驱动包，适用于 IAR 开发环境)

常见问题

YuntulDE添加库

在 YuntulDE 加载完 OSIRE 的示例后，请注意下面几条提示。

右键工程并点击 **Properties** 来打开工程属性窗口。

eclipse-workspace - osine/app/mainc - Eclipse IDE

File Edit Source Refactor Search Project Run Window Help

Project Explorer (in ECLIPSE)

- osine (in ECLIPSE)
 - New
 - Open...
 - Go Into
 - Open in New Window
 - Show In
 - Show in Local Terminal
 - Alt+Shift+W
 - Ctrl+C
 - Ctrl+V
 - Ctrl+X
 - Ctrl+Y
 - Delete
 - Source
 - More...
 - Rename...
 - F2
- Import...
- Export...
- Build Project
- Clear Project
- Refresh
- Close Project
- Close Unrelated Project
- Build Targets
- Index
- Build Configurations
- Profiling Tools
- Run As
- Debug As
- Prints
- Restore from Local History...
- Run C/C++ Code Analysis
- Team
- Create With
- Replace With
- Validate
- Configure
- Source
- Properties
- Alt+Enter

Problems Tasks Console Properties Debug

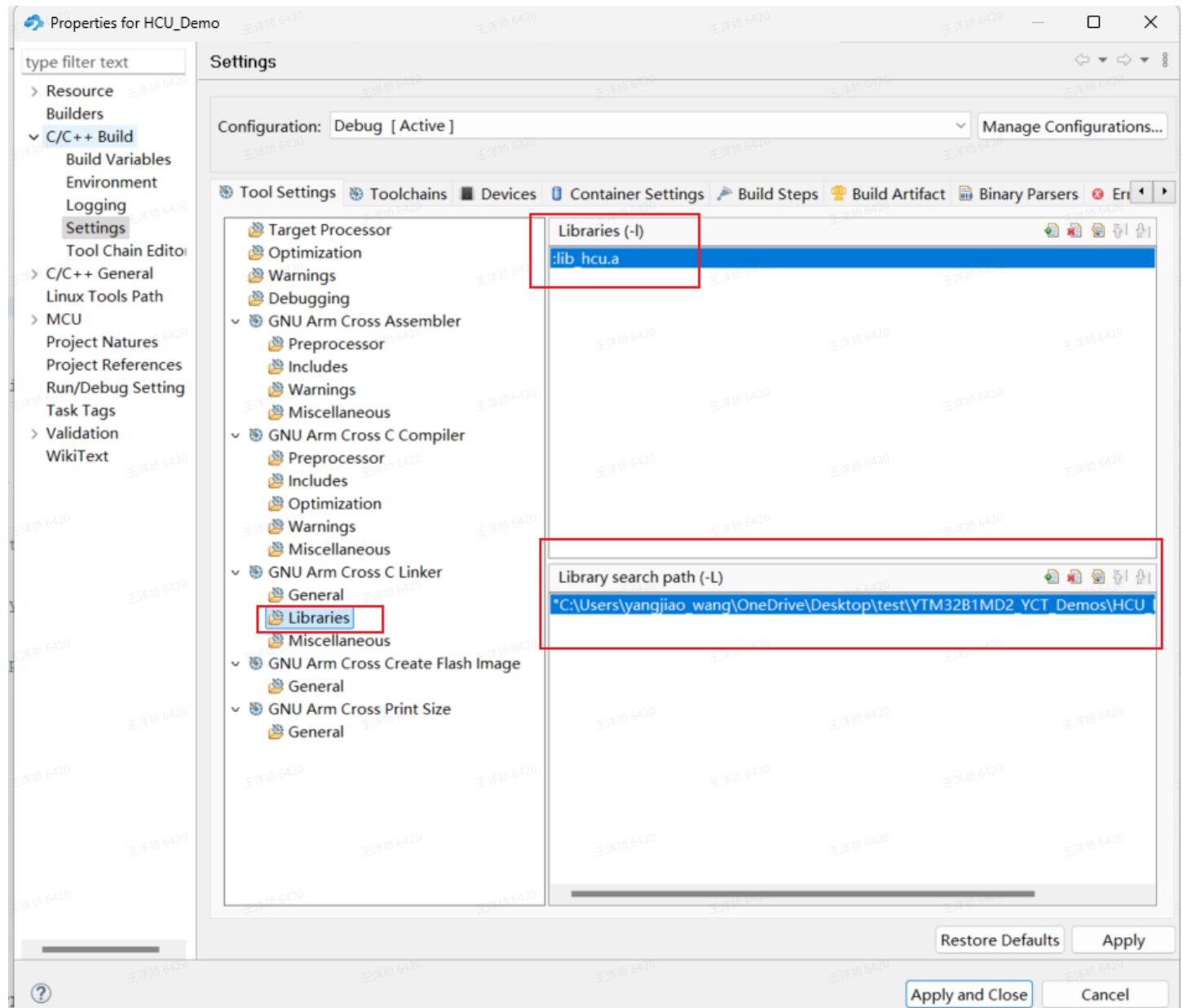
No consoles to display at this time.

```

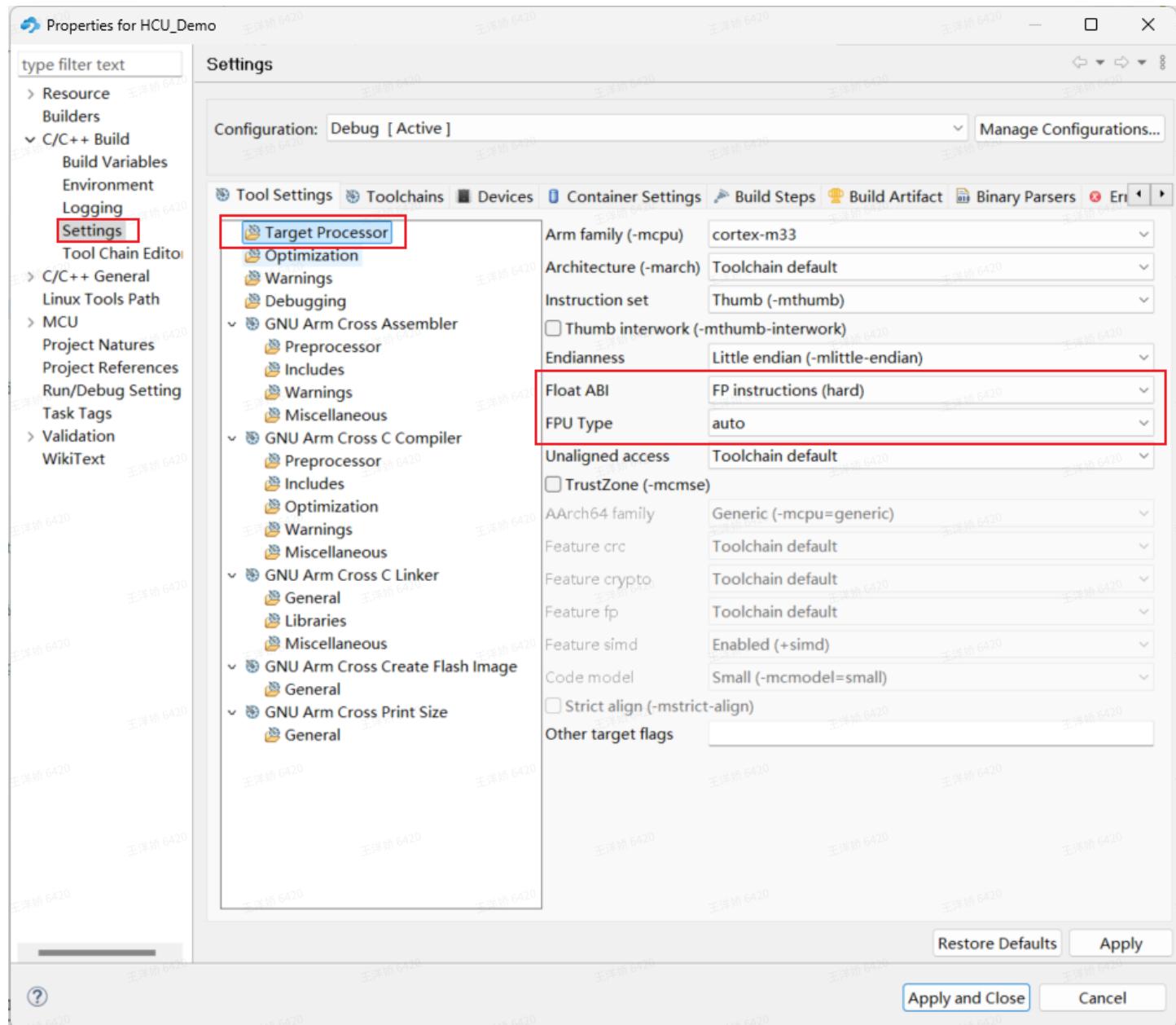
1 // Copyright 2020-2022 Yuntai Microelectronics co.,Ltd
2 // All rights reserved.
3 //
4 // YUNTU Confidential. This software is owned or controlled by YUNTU and may only be
5 // used strictly in accordance with the applicable license terms. By expressly
6 // accepting this software, you agree to be bound by such license terms. If you do not agree to be
7 // bound by such terms, please return this software to YUNTU immediately.
8 // Using this software, you are agreeing that you have read, and that you agree to
9 // comply with and are bound by, such license terms. If you do not agree to be
10 // bound by such terms, please return this software to YUNTU immediately.
11 // Section 2.3 is expressly granted for this software.
12 //
13 // @file main.c
14 // @brief
15 // @version
16 // @date
17 //
18 // @include "adc_driver.h"
19 // @include "peripherals_adc_config.h"
20 // @include "peripherals_i2c_peripheral_config.h"
21 // @include "string.h"
22 //
23 // @define VOLTAGE_DETECT
24 //
25 #include "main.h"
26
27 uint8_t data[8];
28
29 static uint16_t color[6][2];
30 static uint8_t debug = 0x00, ledIndex;
31 static uint16_t color[6][2] = {
32     {0x0000, 0x0fff}, {0xffff, 0x0000}, {0x0fff, 0x0fff}, {0x0fff, 0x0fff},
33     {0x0000, 0x0000}, {0x0000, 0x0fff}
34 };
35 OSINE_PmHandleType dataPm, dataPmReturn, dataPmBlank;
36
37 volatile OSINE_ConnectType appState = IDLE;
38 volatile uint8_t ledIndex;
39 uint16_t ledStatus, ledTemp;
40 uint16_t address = 0, ledCount[1];
41
42 OSINE_FrameType espFrame;
43 void OSINE_CallBack(OSINE_ConnectType state, uint8_t stripNr);
44 void OSINE_FrameStateChange(void);
45 void lightShowMode(void);
46 void OSINE_Calib(OSINE_ConnectType state, uint8_t stripNr)
47 {
48     switch (state)
49     {
50         case TRANSMISSION_COMPLETE:
51             appState = state;
52             break;
53         case RECEPTION_COMPLETE:
54             appState = state;
55             break;
56         case RECEPTION_TIMEOUT:
57             appState = state;
58     }
59 }

```

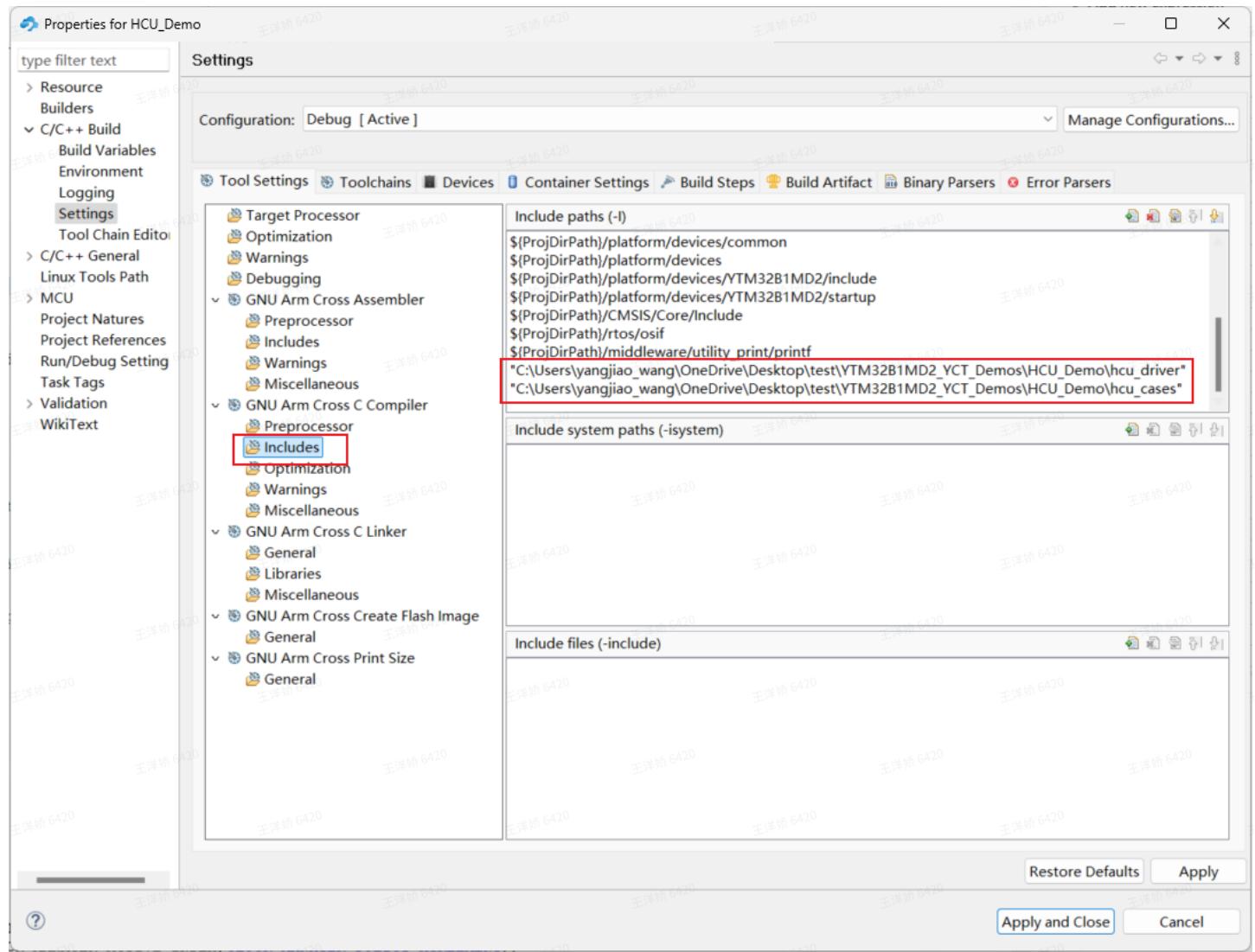
然后点击 **C/C++ Build** 来展开并点击 **Setting**, 并选择 **Libraries**。**Libraries(-l)** 里的内容是实际的库文件, 例如, :lib_hcu.a, **Library search path(-L)** 里的内容是库所在的路径。



配置 FPU 属性，选择硬件 FPU，配置如下



添加路径，增加 hcu_driver 与 hcu_cases 目录



随后正常编译即可。

KEIL添加库

在 KEIL 加载完 HCU 的示例后，请注意下面几条提示。

右键工程并点击 **Manage Project Items** 来打开工程文件配置窗口

C:\Users\yangjiao.wang\work\ym32b1m1\demos\YTM32B1MD1\osire\MDK\osire.uvproj ->Vision

File Edit View Project Flash Debug Peripheral Tools SVCS Window Help

Project: main.c

1 /* Copyright 2020-2022 Yuntu Microelectronics co.,ltd

2 All rights reserved.

3 YNTU Confidential. This software is owned or controlled by YNTU and may only be used and/or reproduced in accordance with the applicable license terms. By express acceptance of such terms or by downloading, installing, activating and/or otherwise using the software, you are agreeing that you have read, and that you agree to the terms of, the applicable license agreement. If you do not agree to be bound by the applicable license terms, then you may not retain, install, private or otherwise use the software. The production use license in

47 package 2.0 is expressly granted for this software.

12 * file: main.c

13 *

14 * Brief

15 *

16 *

17 *

18 #include "mdk_driver.h"

19 #include "peripherals_adc_config.h"

20 #include "mdk_project_config.h"

21 #include <string.h>

22

23 // #define VOLTAGE_DETECT

24

25 uint8_t data[8];

26

27 static uint16_t colorRED[] = {

28 0x0fff, 0x0000, 0xffff, 0xffff, 0xffff,

29 0x0000, 0xffff, 0xffff, 0xffff, 0xffff};

30 static uint16_t colorGREEN[] = {

31 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,

32 0x0000, 0xffff, 0xffff, 0xffff, 0xffff};

33 static uint16_t colorBLUE[] = {

34 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,

35 0xffff, 0xffff, 0xffff, 0xffff, 0xffff};

36 OSINE_PwmDataTypes dataPwm, dataPwmReturn, dataPwmBlank;

37

38 volatile OSINE_CommEventTypes AppState = IDLE;

39 volatile uint8_t stripCallback, ServiceNumber = 0, strip = 0;

40 uint8_t count[5] = {0}, index, address = 0, i, count[3];

41 OSINE_ReturnType state;

42 OSINE_ErrorType error;

43 void OSIRE_Callback(OSIRE_CommEventTypes state, uint8_t stripNr);

44 void FreematicsStateMachine(void);

45 void OSIRE_Callback(OSIRE_CommEventTypes state, uint8_t stripNr)

46 void OSIRE_Callback(OSIRE_CommEventTypes state, uint8_t stripNr)

47 void OSIRE_Callback(OSIRE_CommEventTypes state, uint8_t stripNr)

48 switch (state)

49 {

50 case TRANSMISSION_COMPLETE:

51 AppState = state;

52 break;

53 case RECEPTION_COMPLETE:

54 AppState = state;

55 break;

56 case RECEPTION_TIMEOUT:

57 AppState = state;

58 break;

59 case ERROR:

60 AppState = state;

61 break;

62 case UNKNOWN_ERROR:

63 AppState = state;

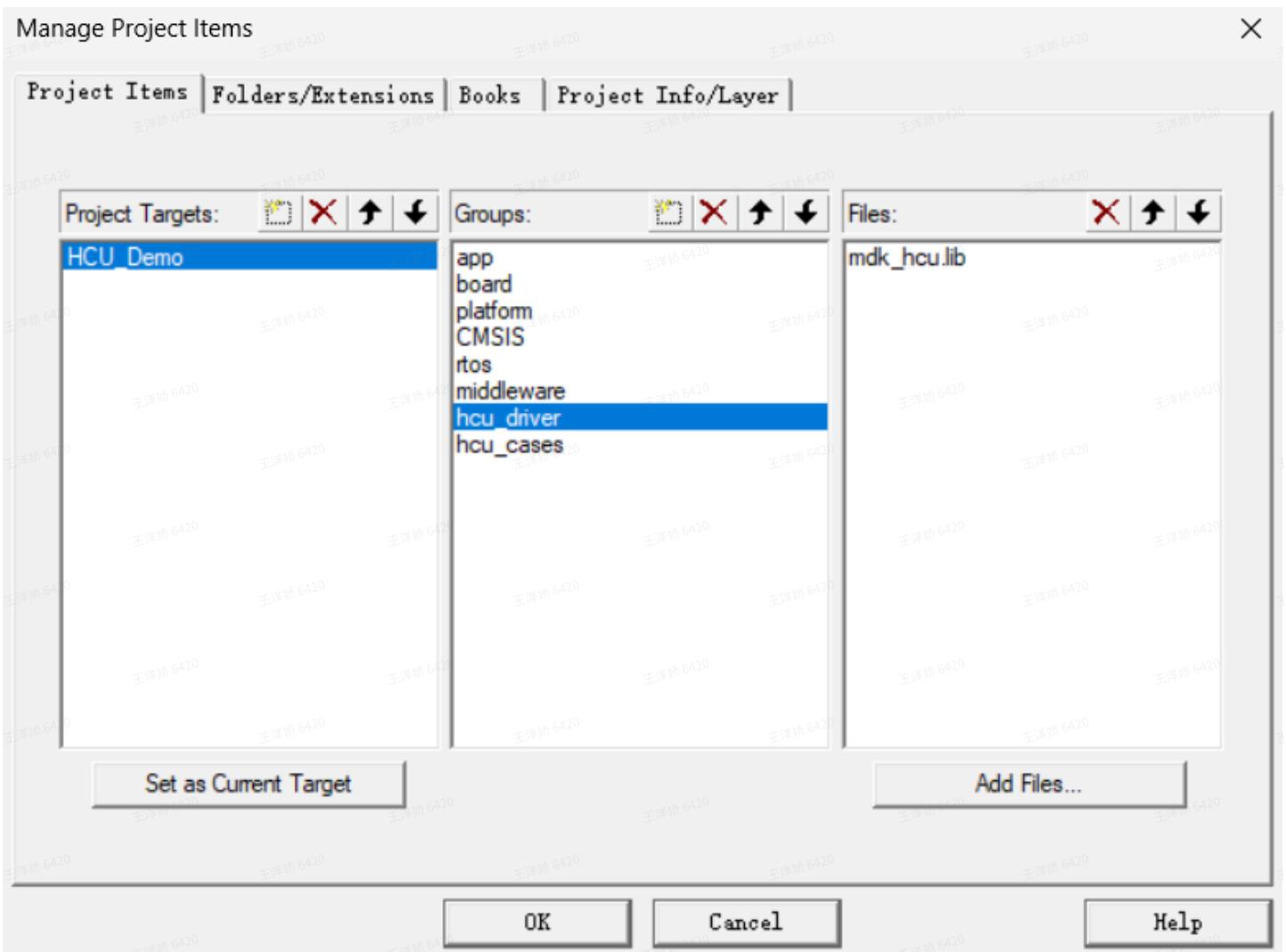
64 break;

65 default:

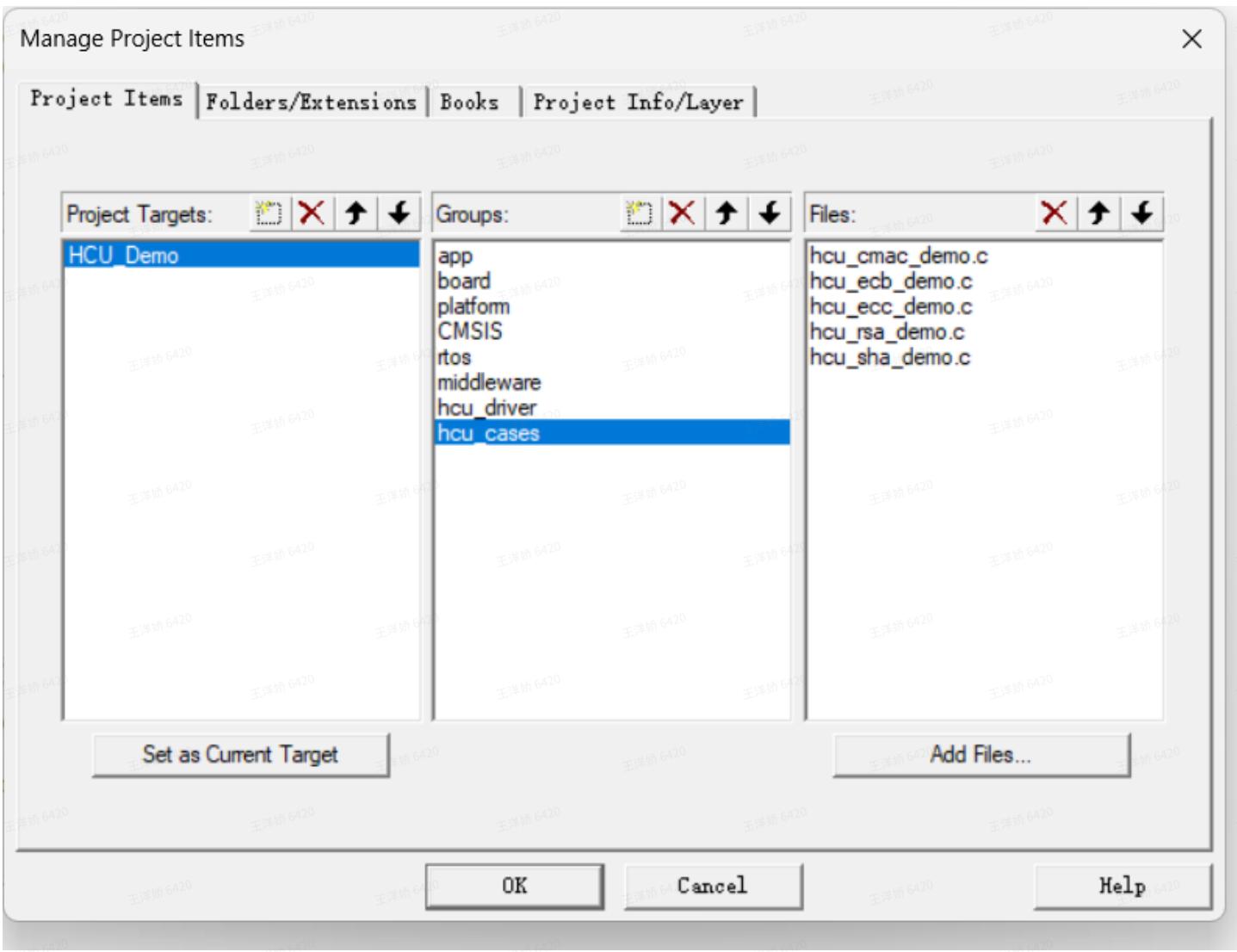
66 }

67

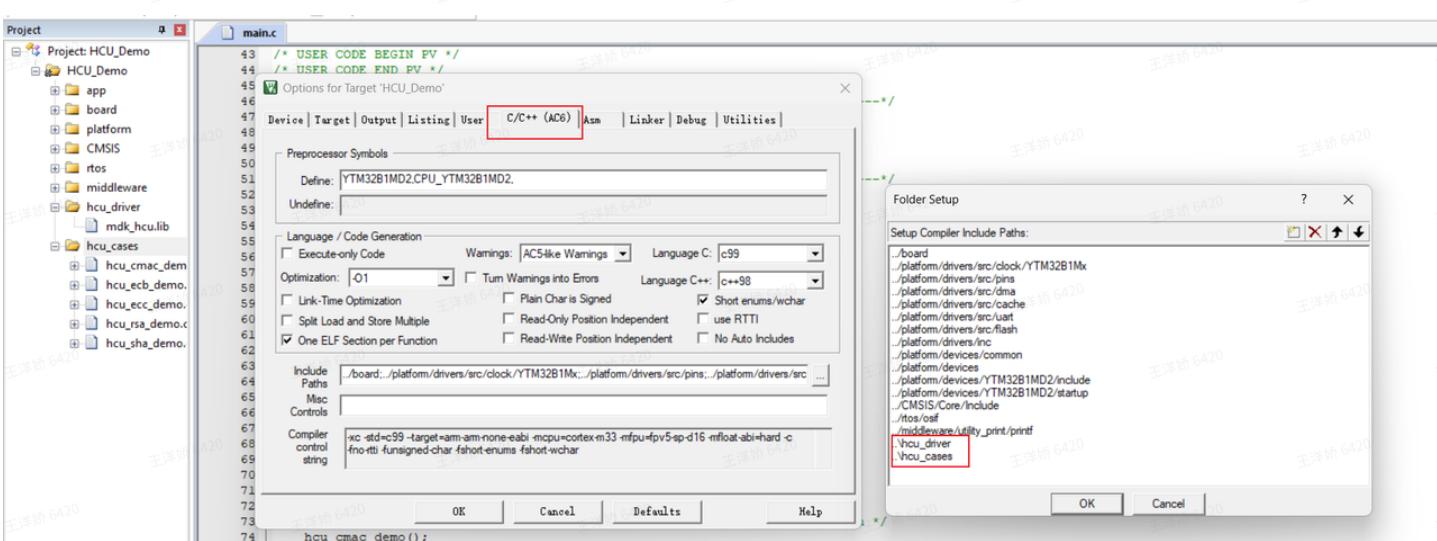
添加 hcu_driver 文件夹，并添加 mdk_hcu.lib



添加 hcu_cases 文件夹，并添加各个例程



添加编译路径，增加 hcu_driver and hcu_cases



随后正常编译即可。

IAR添加库

在 IAR 中加载完 HCU 的示例后，请注意下面几条提示。

右键工程并点击 Add，并选择 Add Group，添加 hcu_driver 的目录与 hcu_cases 的目录。

右键 hcu_driver Group，并选择 Add Files 添加 iar_hcu.a。

右键 hcu_cases Group，并选择 Add Files 添加各个demo例程。

```

/*
 * Copyright 2000-2012 Yuntu Microelectronics co.,ltd
 * All rights reserved.
 *
 * YUNTU Confidential. This software is owned or controlled by YUNTU and may only be
 * used strictly in accordance with the applicable license terms. By expressly
 * accepting this software in its entirety, you accept the license terms and agree to
 * use the software in accordance with those terms. If you do not agree to be
 * bound by the applicable license terms, you may not use this software. Any unauthorized
 * use or otherwise use the software, the production use license in
 * Section 2.3 is expressly granted for this software.
 */

#ifndef _HCU_DRIVER_H_
#define _HCU_DRIVER_H_

#include "peripherals_adc_config.h"
#include "adc.h"
#include "casing.h"

#define VOLTAGE_DETECT
#define ADC_NUM 1

uint8_t data[4];

static uint16_t colorRED[] = {0xffff, 0x0000, 0xffff, 0xffff };
static uint16_t colorGREEN[] = {0x0000, 0xffff, 0x0000, 0xffff, 0xffff, 0xffff };
static uint16_t colorBLUE[] = {0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff };

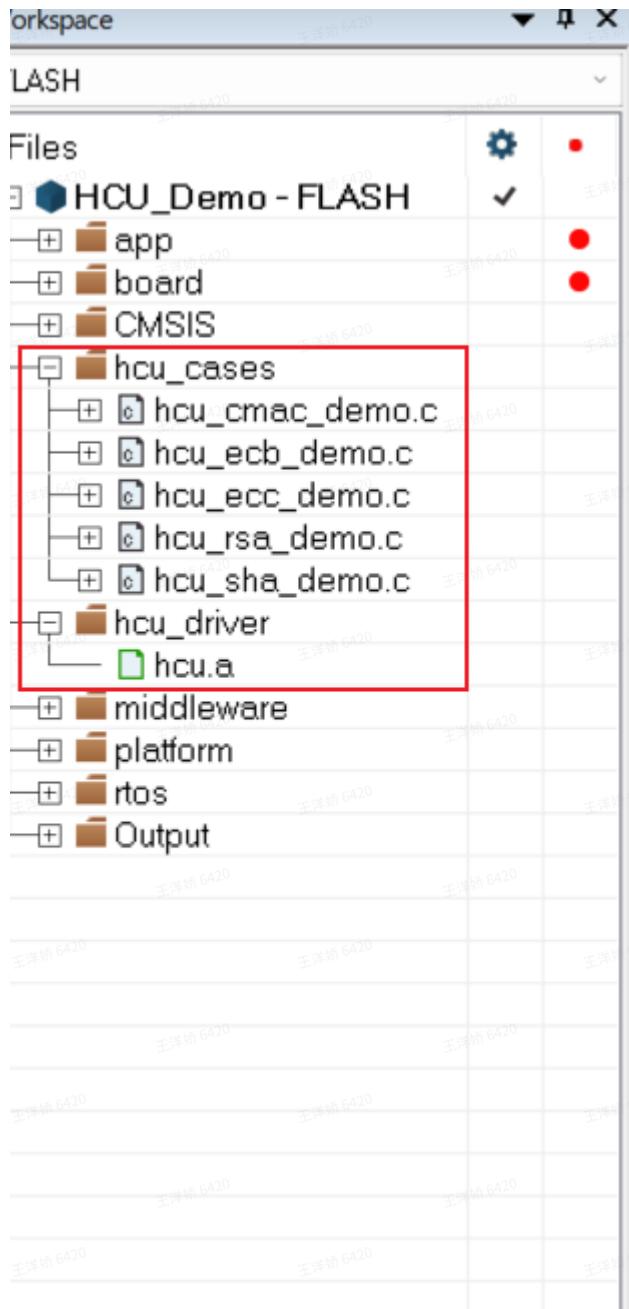
OSIME_PacketReturn database, dataPmblPack;

volatile OSIME_CommandType update = IDLE;
volatile uint32_t stripCallback, ServiceNumber = 0, strip = 0;
uint32_t LedStatus, LedTemp;
uint32_t LedCount[3];
OSIME_ReturnType state;
OSIME_FrameType msgFrame;
void OSIME_CallBack(OSIME_CommandEventType state, uint32_t stripRt);
void FreeseteatStateMachine(void);
void lightShowMode(void);
void OSIME_Called(OSIME_CommandEventType state, uint32_t _stripRt)

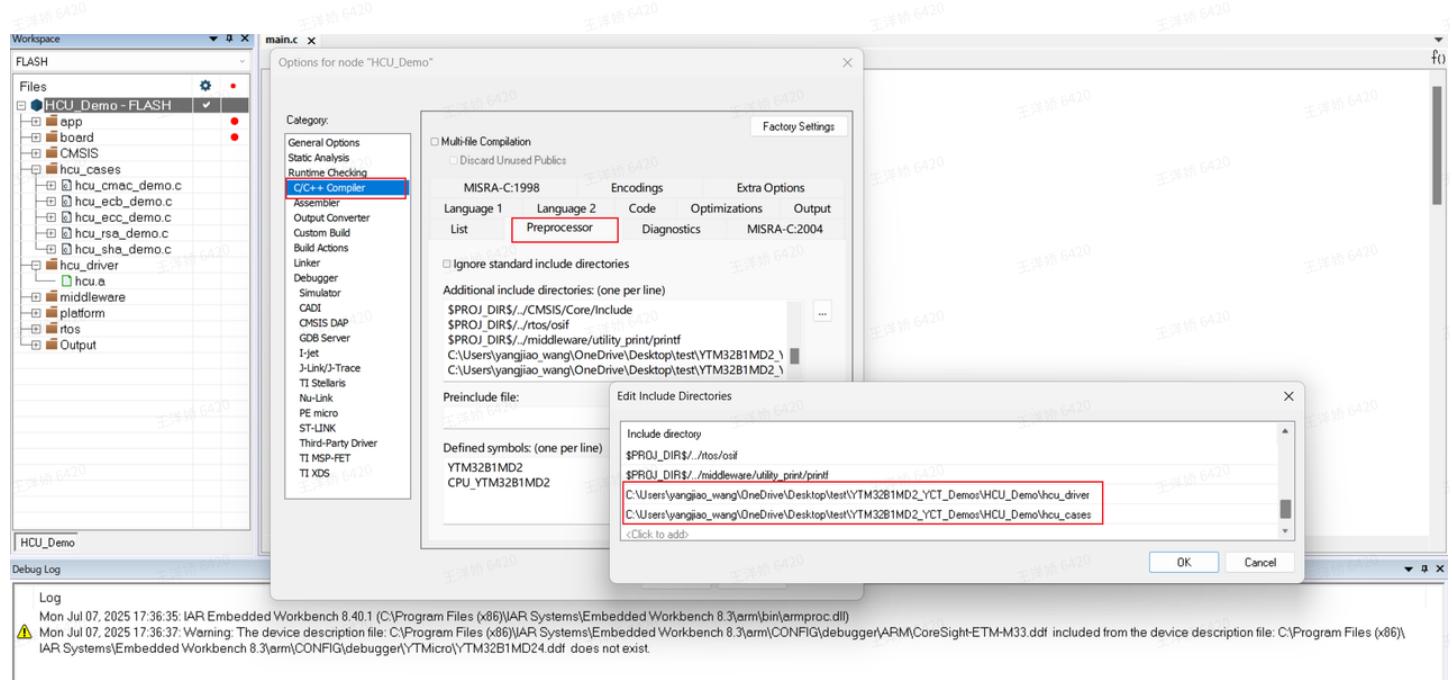
switch (state)
{
    case THRESHOLDISSUE_COMPLETE:
        appState = state;
        break;
    case RECEPTION_COMPLETE:
        appState = state;
        break;
    case RECEPTION_TIMEOUT:
        appState = state;
        break;
    case CRC_ERROR:
        appState = state;
        break;
    case RECEPTION_ERROR:
        appState = state;
        break;
    default:
        break;
}

```

Log
Mon May 29 2023 16:10:57 IAR Embedded Workbench 8.40I (C:\Program Files (x86)\IAR Systems\Embedded Workbench 8.4\arm\bin\mproc.dll)



添加编译路径，增加 hcu_driver and hcu_cases



随后正常编译即可

CMakeGCC

通过使用文件夹内的 CMakeLists.txt , 直接编译。

```
iar_hcu.a M CMakeLists.txt x
HCU_Demo > M CMakeLists.txt
22 # include(...)
23 # USER CODE END include
24
25 set(project_elf HCU_Demo.elf)
26 add_executable(${project_elf} app/main.c)
27 #add app as include path
28 target_include_directories(${project_elf} PRIVATE app)
29 #add all source files in app folder
30 file(GLOB dir_sources "app/*.c" "app/*.cpp" "app/*.S")
31 if(dir_sources)
32   foreach(src ${dir_sources})
33     target_sources(${project_elf} PRIVATE ${src})
34   endforeach()
35 endif()
36
37 # USER CODE BEGIN add_executable
38 # target_include_directories()
39 # target_sources(${project_elf} PRIVATE ..)
40 target_include_directories(${project_elf} PRIVATE "hcu_driver")
41 target_include_directories(${project_elf} PRIVATE "hcu_cases")
42 file(GLOB dir_cases "hcu_cases/*.c")
43 if(dir_sources)
44   foreach(src ${dir_cases})
45     target_sources(${project_elf} PRIVATE ${src})
46   endforeach()
47 endif()
48 # USER CODE END add_executable
49
50 configcore(${project_elf} ${CMAKE_SOURCE_DIR})
51
52
53 # USER CODE BEGIN target_compile_definitions
54 # target_compile_definitions(...)
55 # USER CODE END target_compile_definitions
56
57 target_compile_definitions(${project_elf} PUBLIC
58   "YTM32B1MD2"
59   "CPU_YTM32B1MD2"
60 )
61
62 # USER CODE BEGIN target_compile_options
63 # target_compile_options(...)
64 # USER CODE END target_compile_options
65
66
67 target_link_libraries(${project_elf} "-Wl,--whole-archive" GENERATED_CONFIG_TARGET GENERATED_SDK_TARGET UTILITY_PRINT "-Wl,--no-whole-archive")
68
69 # USER CODE BEGIN target_link_libraries
70 # target_link_libraries(...)
71 target_link_libraries(${project_elf} -L${CMAKE_SOURCE_DIR}/hcu_driver -l_hcu)
72 # USER CODE END target_link_libraries
73
```